Table of Content

Preface

Problem Statement

Consider the network infrastructure needed to sustain an electronic marketplace that is secure, flexible and easy to do business within. Include functions/mechanisms such as authentication, certification, encryption, financial processes and explain how these all fit together.

Contents

This document contains analysis based on the users needs as regards to the problem statement and presents available technological solutions that will meet such needs. The analysis will take into account factors such as price, performance, availability and flexibility.

Organization

An introduction to electronic commerce and the Internet is presented in Chapter 1, which essentially covers the literature review of this project. We take a look into the basis of electronic commerce and then introduce the risks involved. The Internet infrastructure, which forms the foundation of today's connected world is also briefly discussed. We then look at the business requirements in order to make e-commerce feasible on the Internet in the next chapter. Chapter 3 explains in details the two major cryptographic techniques, namely symmetric and asymmetric cryptography, including digital signatures and the Public Key Infrastructure (PKI) that fulfilled the information security need of an electronic marketplace. Functions/mechanisms concerning authentication, certification and encryption are explained in this chapter. We will also see how all these components are pieced together. In Chapter 4 we investigate the real world applications of cryptography and look into the two cryptographic protocols in use today- SSL and SET. We especially outline the details for SET. Finally, a summary is given in the concluding chapter.

Definition of Terms

- Electronic Marketplace

The term *electronic marketplace* refers to the virtual environment (i.e. the Internet) where online retailers (also *e-retailers*) and traders conduct their businesses. The term *electronic commerce* is frequently used in this document to refer to the type of business conducted within this virtual marketplace.

- Users

The word *users* refer to entrepreneurs who plan to set up their online businesses but are in doubt of the feasibility of such investment. Their interests lie in creating a secure and

flexible electronic marketplace where business can be easily conducted. These are the very needs that this document seeks to fulfil.

Abstract

We have seen in recent years the emergence of various techniques and protocols designed to complement and enhance the existing network infrastructure as an effort to fulfill the security need of online operations, especially for electronic commerce. This document addresses various cryptographic applications in authentication, certification, encryption, and financial transaction processes. We look at how all this mechanisms fit together as part of a larger network infrastructure in order to sustain an electronic market place that is secure, flexible and easy to do business within. As regards to the Business-to-Consumers application aspect of these security mechanisms, SET is chosen as the preferred solution to solving the online payment problem over the electronic marketplace. For Business-to-Business electronic commerce, we briefly look at Identrus, which is currently the only available solution in this domain. Next, the feasibility and future of the SET frameworks are discussed.

1 Introduction

1.1 Understanding Electronic Commerce

1.1.1 Background

The term electronic commerce or simply e-commerce can be considered as one of the most ubiquitous business terms in use as mankind embark on the 21st century. As its name suggests, e-commerce requires the digital transmission of transaction information. Here is a definition of e-commerce:

The use of transmission mediums (telecommunications) to engage in the exchange, including buying and selling, of products and services requiring transportation, either physically or digitally, from location to location [15].

The timely arrival of the Internet and the World Wide Web has provided the enabling mechanisms to foster the growth of electronic commerce. Let us begin by understanding the rationale for businesses to establish presence on the Internet, or more specifically, to engage in e-commerce.

1.1.2 Benefits to Sellers

According to [10], Forrester predicts that there will be more than 22 million Internet users by the end of the year 2001. It is also estimated that more than two-thirds of these users will be consumers, and the remaining one-third will be corporate and academic users. The article also quotes Forrester about the sales revenues from online sales that has reached an unequivocal US$ 4.8 billion in 1998 and is expected to reach $1.3 trillion by 2003.

Therefore, the most compelling reason for a business to establish a presence on the Internet is none other than the opportunity that the Internet presents for access to online consumers.

Another factor is globalization. Through the Internet, a business has access to customers in almost any country in the world that has Web access. It is particularly cheap when you consider the alternatives of opening a shop or advertising in foreign countries.

The potential saving in sales costs is another reason to establish business presence on the Internet. It is considerably expensive to establish a shop or a mall, pay the bills, and hire sales staff. By establishing a homepage on the Internet may reduce many of these expenses. These savings can in turn help reduce the costs of the goods and make them more competitive for the businesses.

Additionally, businesses can provide instant updates on their merchandise. Such updates may be particularly useful for selling goods or services that may expire in a very short time. Consumers may obtain these updates instantly on the Internet from the comfort of their home.

Finally, the past few years have witnessed a new trend where digital companies move to just in time inventory (JIT) management. Companies, such as an online bookshop can avoid any inventories and pass on the order from their customers on a just in time basis to the publisher. The publisher then delivers the items to the virtual bookshop, which in return sends the goods to the customers. This new strategy has proved to be successful among software companies [1].

1.1.3 Benefits to Consumers

As most businesses establish their home pages and store fronts on the Internet, we can envision that over time consumers can have access to most, if not all kinds of businesses on the Internet.

Perhaps the most important consumer benefit is the savings in time. Consumers will no longer need to be physically present at a store in order to buy a product, saving them valuable time and energy.

Another reason for a consumer to shop on the Internet is the convenient access to a broad variety of shops and merchandise. This will enable the consumers to compare instantly the quality and price of a given product from different stores and can help make shopping decisions easier and faster.

1.1.4 E-commerce in the Main Stream

Traditional electronic commerce like EDI[1] was rigid and audit-trail oriented, adopted primarily by Business-to-Business transactions and conducted over closed proprietary systems such as industry-specific value-added networks (VANs) rather than over the open domain of the Internet. However, as an increasing number of businesses, individuals, and governments begin to deploy operations and services on the Internet, the way we access and purchase information, communicate and pay for services, and acquire and pay for goods is changing. This newly emerged form of business model is now popularly known as electronic commerce and the virtual environment in which these activities take place is becoming known as the electronic marketplace.

1.1.5 The Risks of Electronic Commerce

Unlike the tightly controlled EDI, today's electronic marketplace is designed to offer access - access to the world instead of a selected few trading partners. However, this presents a conflict between access and control because by opening up access, we give away control. In other words, commerce means access, and risks management means control. The key to balancing between these two conflicts is the key to creating an electronic marketplace that is secure, flexible and easy to conduct business within.

As far as the electronic marketplace in a distributed networked environment is concerned, it is faced with many problems, primarily of security nature. A few of such problems are highlighted below:

[1] Electronic Data Interchange

- The major risk to both clients and server is *eavesdropping* during an electronic transaction. Such trick is accomplished with the installation of small software programs called *packet sniffers* which can be set up to listen to network traffic, looking for interesting patterns in the data (passwords, credit card numbers etc.) at any point along the communication channel. A technically savvy hacker can go so far as altering the contents of the data en route. For instance, a hacker could conceivably modify the content you send to the merchant bank, modifying an electronic fund transfer so that the money is transferred into the perpetrator's account rather than your own.

- Another risk is fraudulent identities. An impostor can trick a Web server into sending him/herself confidential information and vice versa. Therefore, in order to conduct business on the Internet, there must be a way to reliably authenticate individuals and organizations.

- The theft of electronic data such as an organization's customers' financial database is a serious concern to many companies. The only way to defend against such intrusion is by making the Web server as secure as possible. The threat can be reduced by deploying a firewall system which serves as a front-line defense against any unscrupulous attacks. Encryption can also be used as a second line of defense to render stolen data useless.

1.2 Internet

1.2.1 Background

In 1969, the U.S. Department of Defense's Advanced Research Projects Agency (ARPA) sponsored a research project to develop a distributed network that would enable resource sharing among remote users during a war.

The network-ARPANET was designed to transmit fixed-size packets independently through the network. At the originating node, each message was segmented into one or more packets. Each packet carried its own destination address and followed its own route through the network. At the destination node, these packets would be reassembled into a message. This concept has given birth to a protocol that was to become the de facto standard for the Internet. This protocol, called TCP/IP (Transmission Control Protocol/Internet Protocol) is a collection of data communication protocols that allows for easy cross-platform communications. TCP/IP gained wide distribution when it was written into Berkeley Standard Distribution (BSD) Unix, and has since been the standard for connecting different types of computers and networks. Over time, ARPANET, which was used primarily to share information among an elite group of universities and research labs expanded to become the Internet, subsequently causing an information boom that transcends all physical boundaries in the 21st century.

1.2.2 Internet Infrastructure

The past four decades have witnessed the evolution of public telephone network from an all-analogue environment to a virtually all-digital network. Beginning in as early as 1962, the first digital T-carrier was deployed by AT&T, prompting the world-wide telecommunication industry to carry out large scale effort to replace trunks and switches that connect the telephone network to the subscribers with integrated digital networks (IDNs) [2]. Only the local loops leading to the customers' premises (typically known as the last mile) still remain largely analogue.

Until recent years, the public telephone network carries only voice-grade signals. However this changes with the advent of World Wide Web (WWW) that has propelled the growth of digital data and information over the existing public telephone network. Various technologies have emerged to address the need for high-speed communications. Such technologies include improved voice-grade data modems (V.34 and V.90) and ISDN. However, despite the improvement in terms of sustained connection speed, up from the legacy 14.4 kbps to a possible 56 kbps for V.90 modem or 64/128kbps for ISDN, the demand for greater bandwidth increases still. The following new technologies have emerged in recent years to fulfil such a demand.

- ADSL

Bell Communication Research or simply Bellcore (now Telcordia Technologies Inc.) introduced Asymmetric Digital Subscriber Line (ADSL) in 1989 with a technological framework based on the DSL technology [3]. ADSL was developed to leverage the existing base of twisted-pair wire that already links all customers' premises to the public telephone network in order to provide high-speed digital data transmission over ordinary telephone wire. ADSL has since evolved significantly, with the current implementation offering download bit rate from 1.5 to 9 Mbps and as much as 640 kbps downstream [4]. The enormous market potential of ADSL has lead to the formation of the DSL Forum in late 1994 to help telephone companies and their suppliers co-ordinate their development effort. In 1995, ADSL was standardized as ANSI T1.413 and a second issue was made available in 1997 [5].

The DSL technology has also spawn many offspring including:

- SDSL (Symmetric Digital Subscriber Line)
- HDLS (High Speed Digital Subscriber Line)
- IDSL (Integrated Digital Subscriber Line)
- VDSL (Very high Speed Digital Subscriber Line)

All these technologies are similar to ADSL but differ in terms of the modulation techniques employed (e.g. 2B1Q) and variants (either symmetrical or asymmetrical).

- Wireless Technology

Report in [31] found that wireless local loop is fast becoming the favoured option for new entrants to the broadband market. In the UK, the new third generation mobile license is now being auctioned by the government at £1bn each and will enable full Internet services, including video on demand [32]. The mobile Internet is something that has started over the last few years [7], with WAP (Europe,USA), Palm.net (USA) and I-Mode (Japan) making their debuts. In many aspects WAP is similar to normal dial-up access,

but that is about to changed with the proposed introduction of *GPRS*. GPRS can be considered as an upgrade to the existing *GSM* and *TDMA* networks. This means that users are still going to have the same functionality for voice calls, and in addition to that, the benefits of being *always-on*! GPRS can also combined up to 8 time-slots in each time internal for IP-based packet data speeds up to a maximum theoretical rate of 160 kbps [8]. GPRS will likely fuel the migration from the current 2/2.5G mobile network to the latest 3G wireless standard, know an Universal Mode Telephone System (UMTS) [9]. It is believed that the 3G system was driven by Japan's largest cellular carrier NTT DoCoMo with 10 million subscribers. Once deployed in the market, 3G mobile will bring high speed multimedia services (up to 2 Mbps depending on radio condition [9]) to the handset, allowing users to experience many new applications where high-quality voice, text, pictures and interaction converge.

- Satellite

New broadband networks with a global reach are being set up satellite companies, who work closely together with electronic and aerospace corporations to get the satellite up and running. These new broadband networks will connect all the people who live in areas where telephone service over normal copper wire is not available. These networks will be much faster than today's technology at a much lower price, making network access accessible for people with very little income.

- Cable Networks

In the last decades, most industrialized countries have deployed cable network extensively. Today, the existing cable network provides more than just television entertainment, but also high-speed connection to the Internet, through the smart integration of so-called set-top boxes into existing network.

As far as the broadband market is concerned, cable connection s obviously the market leader, with penetration rate up to 90% in the USA [6]. This will likely remain true in the foreseeable future as cable companies can deploy their services much faster and efficient than ADSL provides, which are still laden with the problems of unbundling and interoperability.

- Alternative Technologies

New network infrastructure is deployed to address the phenomenal demand for higher bandwidth. Fiber optic networks, which use optical amplification and photonic switches, are more powerful and more efficient than any existing technology. Telecommunication companies are laying these new fiber cables in an effort to provide a fiber optic Internet backbone. In addition to this software companies have developed new types of compression technologies to reduce the amount of data that is sent over the network.

2 Business Requirements

2.1 Requirements

In the electronic marketplace, the existence of high-speed and efficient Internet infrastructure alone does not address the main problem that electronic commerce faces, which is concerned with information security and online payment. Therefore a good solution incorporating a secure and flexible payment system will lay the foundation towards building trust and confidence among the customers, and make electronic commerce feasible.

2.2 Information Security

The major problem on the electronic marketplace is the identity of its users. In a brick and mortar shop a customer is identified by her or his looks, but on the Internet everyone's identity remains anonymous. This proves to be problematic because during an electronic transaction, the system needs assurance that users are who they say they are. Proving identity is called *authentication*. Usually, one can use a name and password to authenticate a person's identity. Cryptography[2] provides stronger methods of authentication, called digital certificates and signatures. Other possible measures include challenge-response, one-time passwords, smart cards, tokens and biometrics devices [15].

When a message is sent electronically, the send and receiver may desire that the message remain confidential, and thus not be read by any other parties. In order to give an electronic message the property of *confidentiality*, the message must be made unintelligible to everyone except the designated receiver. The principle technique for masking a message is encryption.

The provision for irrefutable proof of the origin, receipt, and contents of an electronic message is called *non-repudiation*. Non-repudiation can be accomplished through the use of bi-directional hashing, digital time stamping, confirmation services or digital signatures.

Finally, apart from keeping order details or credit card information confidential (or encrypted) during transmission, the issue of data integrity is also a major security concern. A message that has not been tampered in any way, either intentionally or unintentionally, is said to have maintained its *integrity*. The cryptographic means of ensuring message integrity is through the use of one-way hashes, also known as *message digest functions*.

[2] The word cryptography stems from the Greek words "kryptos" and "graphos", which literary mean "secret" and "writing" respectively (Bacard, 1995). In plain words, cryptography is the art and science of sending disguised messages so that only the intended recipient of the message can decode it. Cryptanalysis is the science of breaking or analyzing cryptography. Cryptology is the study of both cryptography and cryptoanalysis (Knudsen, 1998).

2.3 Online Financial Transaction

Financial institutions around the world have long established their own private payment infrastructure (SWIFT, CHIPS, CHAPS, Fedwire etc.) to facilitate financial transactions in banking [16], [1]. Unfortunately, electronic payments systems over the Internet, due to its open nature are much harder to regulate and control. The Internet allows anybody to eavesdrop on the traffic, so modification of messages needs to be prevented by the use of digital signatures and encryption technologies. Financial transactions could nonetheless be performed via the Internet. For every transaction, it involves a buyer and a seller (or merchant). In order to perform a financial transaction a financial institution is required that enables the money transfer. In most cases two financial institutions are involved. The *issuer* is the financial institution used by the buyer and the *acquirer* is used by the merchant. Electronic payment starts with the communication between buyer and issuer, whereby the buyer asks the issuer to release money by withdrawing it from a bank account. The money is then sent to the acquirer for clearing. If the acquirer validates the money, a message will be sent on to the merchant. The merchant can then start the order processing with the knowledge that the money has been put into the merchant's account. However, this model still lacks the information security needed to emulate the properties of existing payment schemes.

Online payment infrastructure needs to be flexible. They should support different kinds of payment methods such as post-paid (e.g. credit card), pre-paid (e.g. DigiCash, Electronic Cash), and instant-paid (debit card). Currently, the most widely used payment solution is credit card. However, credit card transactions via SSL, which is the most widely deployed security protocol on the Internet, do not guarantee that the transaction/payment can be fulfilled successfully. Even if money has been transferred to a merchant's account, the cardholders are able to enforce a charge back [1]. This is because Internet credit card transactions are classified as Card-Not-Present transactions, therefore merchants are liable for losses, even when the bank has authorized the transactions. There is a need to address this type of fraudulent transaction that could result in loss of revenue.

In order to make the online payment infrastructure successful, the requirements as underlined in the Information Security section must be fulfilled. Additionally, it should be easy to use and widely accepted. The payment solution should also be an open standard to ensure transparency and flexibility. An appealing solution to customers should handle most of the payments automatically and customers do not need to understand the structure behind the payment mechanism.

3 Concepts

3.1 Cryptographic Techniques

As discussed earlier, cryptography is used primarily to address the four major security requirements - confidentiality, authenticity, integrity and non-repudiation - involving data transmission over the Internet.

The primary method of achieving confidentiality is encryption [11]. The elements of an encryption system are the *plaintext* (or *cleartext*), the cryptographic algorithm, the key, and the *ciphertext*. The plaintext is the actual message or data that is to be encrypted (scrambled). The cryptographic algorithm or *cipher* is a mathematical set of rules that defines how the plaintext is to be combined with a key. The key is a string of digits, and the ciphertext is the resulting encrypted message.

These terms can be shown with a very simple example, called the Caesar Cipher. Let us take the phrase “encryption” and add 3 characters to each letter, the phrase becomes “hqfubswlrq”.

In this situation:

“Encryption” is the plaintext

“add 3 characters to each letter” is the cryptographic algorithm or cipher

“3” is the key

“hqfubswlrq” is the ciphertext

There are many other encryption techniques, including classical ones like the substitution ciphers consisting of monoalphabetic, homphonic, polyalphabetic and polygram, transposition techniques, Playfair Cipher, Rotor cipher and Hill Cipher [22]. All rotors ciphers are polyalphabetic substitution ciphers and were the most important cryptographic devices used in World War II, and remained dominant until the introduction of DES.

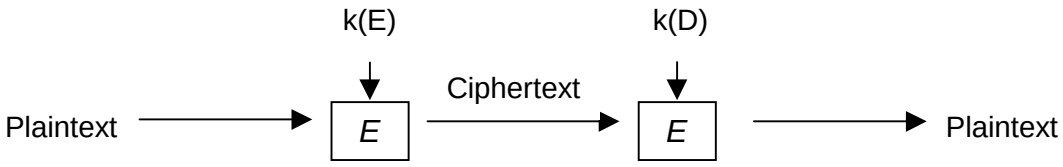

Figure 1: The encryption and decryption processes

The processes of encryption and decryption can also be illustrated in Figure 1 where the encryption key is denoted by k(E), whilst the decryption key is denoted by k(D). *E* represents the encrypted message (ciphertext). In order to maintain the privacy of the

message, k(D) must remain secret, known by as few people as possible, preferably only by the recipient of the message. There are two types of cryptographic algorithm that can be used to provide encryption capabilities: symmetric cryptography and asymmetric cryptography (also known as public key cryptography). A symmetric cipher is one in which the keys k(E) and k(D) are such that knowledge of k(E) implies knowledge of k(D). In this case, therefore, it is necessary for k(E) to be secret also. An asymmetric cipher is one in which it is practically impossible (i.e. computationally infeasible) to obtain k(D) from knowledge of k(E). Here, k(E) need not be kept secret. Both symmetric and asymmetric cryptographic techniques are discussed in the next sections.

3.2 Symmetric Cryptography

3.2.1 Data Encryption Standard

The primary symmetric encryption algorithm widely used in the computer industry today is the Data Encryption Standard (DES). IBM developed this technique in 1971 as part of a research project designated as LUCIFER [12]. In 1977, the U.S. government adopted DES as the national cipher standard [12].

DES is a block cipher based encryption technique that takes a 64-bit fixed block of text, and transforms the 64-bit block of text into a 64-bit block of ciphertext. In DES symmetric key systems, both the sender and receiver of the message must have access to the same key. This shared key is also 64-bit long, of which only 56-bit is used to encrypt and decrypt the message. The remaining 8-bit is used for parity checks. The total number of keys is thus $2^{56} = 7.2 \times 10^{16}$.

The DES key algorithm essentially consists of a series of permutations and substitutions. A block that is going to be encrypted is first subjected to an initial permutation (*IP*), then to a complex series of key-dependent operations and finally to another permutation (IP^{-1}), which is the inverse of the initial permutation.

3.2.2 Other Symmetrical Algorithms

Other common symmetrical algorithms likely to be used in the computer industry are Triple DES, AES, RC2, RC4, RC5, IDEA and Blowfish.

- Triple DES

Triple DES or 3DES is a variant of DES that decreases the risk from brute-force attack by using longer keys. Other similar variants include DESX, GDES and RDES. In the widely used triple-DES, the message is first encrypted with one secret key, next decrypted with a second secret key, and finally encrypted again with the first secret key. The message is decrypted by reversing the procedure. This gives an effective key length of roughly 168-bit.

- RC2, RC4 and RC5

These are proprietary algorithms invented by Ron Rivest from RSA Data Security, Inc for very fast bulk encryption. They are alternatives to DES and are as fast or faster than

DES. They can be more secure than DES because of their ability to use long key sizes; they can also be less secure than DES if short key sizes are used.

RC2 is a variable-key-size symmetric block cipher and can serve as a drop-in replacement for DES, for example in export versions of products otherwise using DES. RC2 can be used in the same modes as DES including triple encryption. RC2 is approximately twice as fast as DES, at least in software. RC4 is a variable-key-size symmetric stream cipher and is 10 or more times as fast as DES in software. Both RC2 and RC4 are very compact in terms of code size. RC5 is also a symmetric block cipher like RC2 and has variable block sizes ranging from 32 to 128 bits and a variable length ranging as high as 2,048-bit.

RC2 and RC4 are widely used on the Internet as crippled versions that use 40-bit keys instead of the stronger 56-bit or 128-bit. The export of only crippled versions of RC2 and RC4 algorithm is due to restriction imposed by the U.S. government which treats such cryptographic algorithms as munitions, along with certain missiles and nuclear arms. However, this restriction does not affect the export of such cryptographic algorithms in other countries.

- AES

In 1997, the NIST[3] of the government of the United States announced the initiation of a process to develop the AES (Advanced Encryption Standard) and serve as an eventual successor to the venerable DES [13]. NIST subsequently published its formal call for algorithms to the world wide cryptographic research community. After four years of grueling evaluation, the candidate algorithm RIJNDAEL has finally been chosen as the new AES. RIJNDAEL is based on a powerful substitution-linear transformation network with ten, twelve, or fourteen rounds, depending on the key size, and with variable block sizes of 123-256 bits.

- IDEA

The International Data Encryption Algorithm is a symmetric block cipher developed by Xuejia Lai and James Massey of the Swiss Federal Institute of Technology (Stallings, 1999). IDEA is one of a number of conventional encryption algorithms that have been proposed in recent years to replace DES. IDEA is a block cipher that uses a 128-bit key to encrypt data in block of 64 bits. By contrast, DES also uses a 64-bit blocks but only a 56-bit key.

3.2.3 Key Length and Security for Symmetric Cryptography

The strength of an encrypted message is depended upon the length of the encryption key. Given a good encryption algorithm, the longer the key length, the more secure the message. The only way to crack an encrypted message is to try all possible values of the key until the correct one is found. This brute-force approach will take time and money. As the key size increases, so does the cost and time of a brute-force attack. Every bit doubles the number of possible keys. A 57-bit key will take trice as long to crack as a 56-bit key. Therefore, a reasonable approach is to choose a length whose

[3] National Institute of Standards and Technology

cost to crack exceeds the value of the encrypted message by some comfortable margin. No person will speed £1,000,000 to crack a message whose value is only £10,000.

The calculation of comfortable key length should also take Moore's Law into account. This empirical law observes that computers double in speed every 18 months, effectively decreasing the cost of cracking a message by a factor of 10 every five year or so. In accessing key length security, we need to weigh the value of the message against the cost of cracking it and the expected longevity of the message.

The length of the key should also be balanced against performance. The longer the key length, the more time is consumed to encrypt/decrypt a message. The security of various key lengths given today's computational power is shown in Table 1 [14].

Key length	Time
40-bit	Seconds
56-bit	Hours
64-bit	Days
80-bit	Millennia
128-bit	> Age of the universe

Table 1: Estimated time to crack symmetric algorithms with various key lengths

3.2.4 Weaknesses in DES

The challenge with DES lies in the proper dissemination of the secret key between the sender and receiver, especially in those instances when the sender and receiver are geographically separated. As a network proliferates, the secure exchange of secret keys becomes increasingly expensive and unwieldy. Consequently, this solution alone is impractical for moderately large network.

Fundamental issues that DES does not address are authentication and non-repudiation. DES does not provide for verification of the origin or identity of the sender, so there is no authentication. Senders can falsely deny sending or receiving the message thus rendering DES useless in terms of non-repudiation. The needs for proof of origin, receipt, and contents can only be achieved through public-key cryptography. For most of its life, the primary concern with DES has been its vulnerability to brute-force attack. This was initially attributed to its relatively short key length (56-bit). However, with the increasing popularity of block ciphers with longer key lengths, including triple DES (168-bit), brute-force attacks have become increasingly impractical. In recent years, we have seen the emergence of cryptanalytic attacks on DES and other symmetric block ciphers notwithstanding. The two cryptanalytic approaches that have assumed importance are *differential cryptanalysis* and *linear cryptanalysis*.

3.3 Asymmetric Cryptography - Public Key Cryptography

The problem of key distribution with the conventional symmetric system prompted Whitfield Diffie and Martin Hellman from the University of Stanford to come up with a different approach, named the asymmetric cryptographic technique as an effort to achieve the security objectives of cryptography. The asymmetric technique provides a radical departure from conventional cryptographic methods of substitution and permutation with the manipulations of very large prime numbers. This concept was adopted by Ron Rivest, Adi Shamir and Len Adlemena at MIT and has since been further developed to become the widely popular RSA algorithm, one of the few asymmetric or public key (as it is generally known) algorithms currently in use today. Other algorithms include the Diffie-Hellman Key Exchange and Digital Signature Algorithm (DSA. A model for public key cryptography is illustrated below in Figure 2.

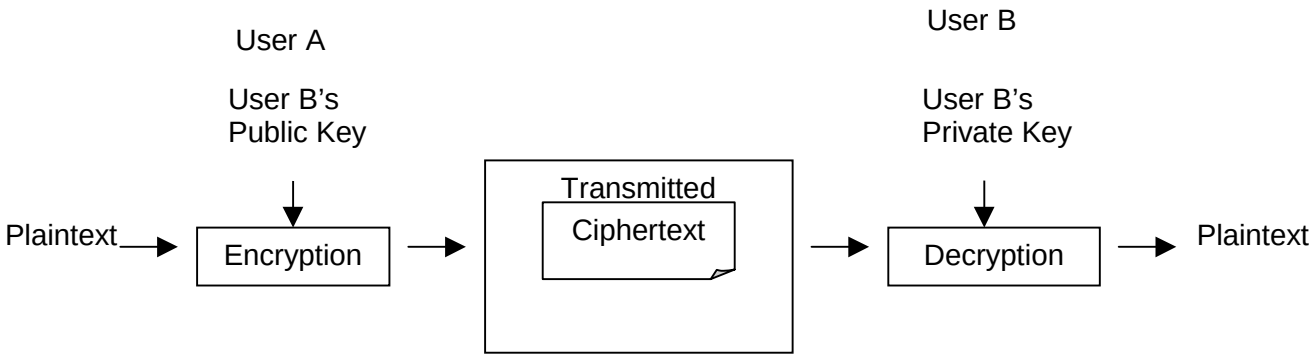

Figure 2: The public key cryptosystem

The public key cryptosystem (PKCS) is asymmetric involving the use of two separate keys, in contrast to symmetric system, which uses only one session key. The two keys used for public key cryptography are referred to as the *public key* and the *private key*, known simply as the key pair. The key pair can be used for encryption, decryption or both, which can provide message authentication and confidentiality.

For example, User A wants to send an encrypted message to User B. First, User A obtains the public key of User B, which is available freely through the Public Key Infrastructure (PKI), discussed later. The public key of User B is used to encrypt the message that User A plans to send. The encoded message is then transmitted and eventually reaches the recipient, which is User B. User B then uses his/her own private key to decrypt the encoded message. Both parties can be rest-assured of the message confidentiality because only the person who is in possession of the matching private key can convert the encoded message back into cleartext form.

The scenario given above, however, does not address the issue of message authentication. Anyone can encode a message with User B's public key and impost as User A. This is a problem where the use of digital signature can resolve (discussed in this chapter).

3.3.1 RSA Algorithm

The RSA algorithm works by generating two large primes, *p* and *q*, and calculate their product *n*: *n = p x q*; n is called the modulus. Then a number *e* is determined, which is less than *n* and relatively prime to *(p – 1)(q - 1)*. In other words, the largest common factor of *e* and *(p – 1)(q - 1)* is 1. The number *e* also satisfies the following expression:

$3 < e < (p - 1)(q - 1)$

The same number, *e*, is then used to determine its inverse, *d*, for which

$ed = 1 \pmod{(p - 1)(q - 1)}$

Here, *e* and *d* are called the public and private exponents respectively. The public key is the pair (n,e) whereas the secretly kept private key is *d*. The factors *p* and *q* must be kept secret, or destroyed. It is difficult (presumably) to obtain the private key *d* from the public key (*n*,*e*). If one could factor *n* into *p* and *q*, however, then one could obtain the private key *d*. Thus the entire security of RSA is predicated on the assumption that factoring is difficult. There is currently no available easy factoring method that could break the RSA algorithm. The applications of RSA can provide for privacy and authentication (discussed in section 3.4- Parallel Use of Symmetrical and Asymmetrical Techniques) in Internet communication.

3.3.2 RSA Encryption

Suppose A wants to send a private message, *M*, to B. A creates the ciphertext, *C* through the following exponentiation function:

$C = M^{e} \pmod{n}$

where *e* and *n* are B's public key. In order to decrypt the ciphertext, B performs the next exponentiation function:

$M = C^{d} \pmod{n}$

Since only B has knowledge of *d*, B will be able to recover the original message *M* using the above function. The relationship between *e* and *d* ensures that B correctly recovers *M.*

3.3.3 RSA Authentication

Suppose A wants to send a signed document *M* to B. A creates a digital signature, *S* by exponentiating

$S = M^{d} \pmod{n}$

where *d* and *n* belong to A's private key pair. She sends both *S* and *M* to B. To verify the signature, B exponentiates and verify that the message *M* is correctly recovered:

$M = S^{e} \pmod{n}$

where *e* and *n* belong to A's public key. Thus encryption and authentication take place without any sharing of private keys: each person uses only other people's public keys and his or her own private key. Anyone can send an encrypted message or verify a signed message, using only public keys, but only someone in possession of the correct private key can decrypt or sign a message

3.3.4 Key Length and Security

It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key, a feature that makes public key cryptography so unique. In addition, some algorithm, such as RSA, also allows for either of the key-pair to be used for encryption, while the other used for decryption. The RSA algorithm has essentially paved the way for digital signatures.

For technical reasons, the keys used by public key algorithms need to be larger than symmetric keys in order to provide the same level of resistance to brute-force attacks. A 384-bit public key pair provides approximately the same level of security as a 40-bit symmetric key [17]. Unlike symmetric keys, which are typically discarded after a single communication session, public key pairs have to remain secure for a long time. A CA's public certificate, for instance, may be used to encrypt millions of session keys during its lifetime. For this reason, even 512-bit keys are considered to be low security. Most companies, especially in the United States, use keys of 1,024-bit or higher. Table 2 shows the public-key lengths equivalence of symmetric key length in terms of security.

Symmetric Key Length	Public Key Length
56-bit	384-bit
64-bit	512-bit
80-bit	768-bit
112-bit	1,792-bit
128-bit	2,304-bit

Table 2: Symmetric key and Public Key Lengths for Equivalent Levels of Security

3.3.5 Weaknesses in Public Key Cryptography

The problem with RSA cryptography is that it is not as efficient as the DES symmetric method. DES is much faster than RSA public key cryptography by a margin of 100 times in software, and up to 1,000 times in hardware.

In terms of the security of RSA algorithm, the algorithm is prone to three possible types of attacks- brute force attacks, mathematical attacks and timing attacks. The second type of attacks, namely the mathematical attacks occupy the center stage in modern RSA cryptanalysis because of the so-called factoring problem.

3.4 Digital Signature and Message Digest

The development of digital signature stems from the concept that messages encrypted using an individual's private key can only be decoded with the matching public key. Digital signature is clever reversal of the public key encryption/decryption scheme. It provides a simple way for creating an irrefutable evidence of the owner's identity and, in some cases, authority in a given transaction. The following two possible situations can be of utmost legitimate concern:

- User A may forge or alter a message, append an authentication code using the key both of them share, and claim that it came from User B. User B might not be able to deny such an allegation given the little evidence there is to support his testimony. For example, an online dealer can increase the number of order placed for a certain product between him and a consistent purchaser and claim that the orders have come from the purchaser.

- User B may deny sending a message to User A, which User B *did* in fact send. There is no way User A can prove that User B is the author of the message because User B may argue that it is possible for User A to forge the message. For instance, a buyer sends instructions to a stockbroker for a transaction that turns out badly. The sender can pretend that the message was never sent.

Any public key algorithm can be used to create digital signatures, however the RSA algorithm is the most popular. There is also an algorithm that was specifically developed for this purpose, called Digital Signature Algorithm (DSA).

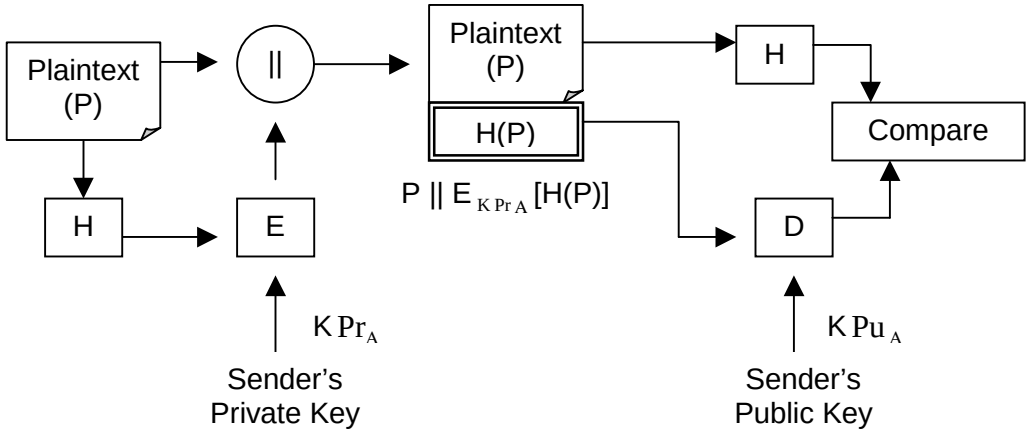

Figure 3: The Essence of Digital Signature Implementation

Figure 3 shows the essence of digital signature implementation. First, the sender hashes (as denoted by *H*) the plaintext and produces an integrity check value known as *message digest*. A hash is a one-way function and is often referred to as a message digest algorithm. The message digest produced is much smaller than the original message in format therefore making it an efficient way to represent the message, as well as being a unique number that can only be calculated from the contents of the message. Also, collisions (two sets of data that produce the same hash) and forgery are computationally infeasible.

The hash is computed with a hashing algorithm that is not secret. No key is used. There exists an alternative method, called the Message Authentication Code (MAC) that combines a key with a hash. Hashing algorithm and MAC are used in such a way that if even a single character in the message is altered, a different message digest altogether will result. Obtaining the full set of data from the hash is also virtually impossible.

At this point, the sender's private key is used to encrypt the message digest. The encrypted message digest is what is commonly referred to as the digital signatures. It is more efficient to compute a digital signature based on the message's digest, which is small, rather than using the arbitrarily large message itself because the hash functions are faster than the digital signing functions (e.g. RSA algorithm).

The digital signature is a unique creation of the contents of the message and the sender's private key. Therefore, the message can be attributed to no one else but the indisputable author. The message and the appended digital signature are transmitted to the recipient. The recipient recalculates the message digest using the agreed upon hashing algorithm. Then, the recipient uses the sender's public key to decrypt the message digest. If the recalculated message digest is identical to the decrypted message digest, then the message can be attributed to the sender (authenticated) and its contents considered free from tampering (integrity-checked).

3.4.1 Digital Certificates

Digital certificates, also know as digital IDs, bind an identity of an individual or an organization to a pair of electronic keys that can be used to encrypt and sign digital information. A digital certificates makes it possible to verify someone's claim that they have the right to use a given key, helping to prevent people from using phony keys to impersonate other users.

Owner's identifying information: name, organization, address etc.
Owner's public key
Issuing authority's identifying information and digital signature
Expiration data of this digital certificate
Other information: class of certificate etc.
Certificate's serial number

Figure 4: General Contents of a Digital Certificate

In their simplest form, digital certificates contain an owner's name and its public key. As commonly used, they also contain the expiration date of the certificate, the name of the certifying authority that issued the certificates, the serial number of the certificate, and possibly other information. Most importantly, a certificate contains the digital signature of the certificate issuer (Figure 4). The most widely accepted format for certificates is defined by the CCITT[4] X.509 international standard (now ITU-T[5] X.509), which is further discussed in the PKI subsection. ISO[6] is currently developing a series of certificate management standards knows as ISO 15782. These standards may replace ITU-T X.509.

Public key cryptography requires access to users' public keys. In a large-scale networked environment it is impossible to guarantee that prior relationships between communicating entities have been established or that a trusted repository exists with all used public keys. Therefore, digital certificates were invented as a means to validate the secure distribution of public keys, with which the exchange of symmetric keys is carried out through the Internet, thus ensuring message authentication and confidentiality. The validation process is done by comparing the calculated hash value based on the owner's public key and name with the decrypted hash value of identical entities previously signed by the issuing CA (Figure 5).

[4] Comité Consultatif International de Télégraphique et Téléphonique (CCITT)
[5] International Telecommunications Union
[6] International Organization for Standardization (ISO)

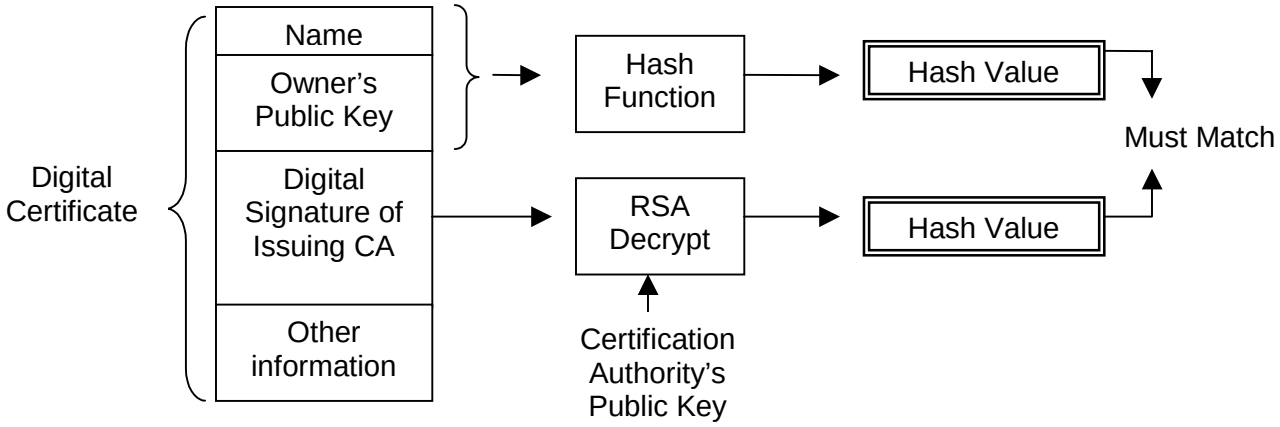

Figure 5: Digital Certificate Validation Process

Used in conjunction with encryption (DES key) and hash functions (Digital Signature), like the RSA public key cryptosystem, digital certificates provide a more complete security solution, assuring the integrity and secrecy of its message, and identities of all parties involved in a transaction.

3.4.2 Differences between Digital Certificates and Digital Signatures

Both digital certificates and digital signatures are based on the same public-private keys pair concept. In fact, digital signature is simply a subset of the whole digital certificate schema. When a person applies for a digital certificate from a Certification Authority, which is a trusted body that manages the distribution of public keys, a *key pair* consisting of the owner's *public* and *private* key will be generated. It is vital to the overall security that the private key remains known only to whom it belongs (either a person or server). Additionally, the owner has the discretion to distribute the public key to his/her correspondents.

The private key is the entity that produces digital signatures that are used to validate message integrity and authentication (refer to Figure 3). If both communicating parties employ digital signatures, the objective of non-repudiation can be accomplished.

3.5 Public Key Infrastructure and Certification Authority

The secure distribution of both the public and private keys are to be tightly controlled because these keys are bound to each individual and therefore is susceptible to forgery and masquerading. The distribution of keys, usually in the forms of digital certificates is accomplished through a central authority, which maintains a dynamic directory of public keys of all subscribers. This central authority is generally regarded as the Certification Authority (CA) and can be any trusted party willing to vouch for the identities of those to whom it issues certificates. A CA will accept a person's public key, along with some

proof of the person's identity (it varies with the class of the certificate), and serve as the repository of digital certificates that others can request to verify a person's public key. Apart from publication and distribution of certificates, the repository also contains revocation information for invalid certificates. All these components form the backbone of an entity we regard as the Public Key Infrastructure (PKI).

3.5.1 Root CAs and Certificate Chains

Most browsers, including Netscape Navigator and Internet Explorer are shipped with the public keys of several certifying authorities preinstalled. The public keys are installed in the form of *self-signed* certificates, which are digital certificates that the CA itself has signed. These trusted CAs are also known as *root CAs* and includes such companies as Verisign, Thawte, Entrust and so on. Netscape Navigator Version 4.61 recognizes about 20 such CAs whereas Internet Explorer Version 5.0 recognizes approximately 5.

In addition to signing end users' certificates, a root CA can sign another CA's public key, granting it signing authority. This starts a certificate chain. The secondary CA can now sign end-user certificates or sign the certificates of CAs further down the chain. When a user presents a certificate signed by one of these nonroot CAs, the receiver will have to validate all the CAs in a hierarchy between the sender's local CA and the issuing CA. That could include traveling up one branch of a CA hierarchy to the root before the sender's public key can be recovered safely. This is called a *hierarchy of trust*.

3.5.2 Acquisition of Certificates

There are three approaches to generating public-private key pairs and digital certificates, which are:

1. The user generates his/her own key pair. This method has the advantage that a user's private key is never released to another entity, only the public key and other relevant user information contained in a *certificate request* is sent to a CA. The CA will produce and sign the certificate based on the information given. The key pair can be generated using a browser or tools available on most server platforms, such as Windows NT running IIS and C2Net Stronghold.

2. The key pair is generated by a third party. The third party shall release the private key to the user in a physically secure manner, then actively destroy all information relating to the creation of the key pair plus the keys themselves. Suitable physical security measures shall be employed to ensure that the third party and the data operations are free from tampering.

3. CA generates both the key pair and certificate before sending them to the subscriber. Key pair data stored on a local machine is prone to illegal copying and theft, so a suitable method for storing it in a convenient transportable manner should be considered. One possible mechanism would be to use a *Smart Card*. The smart card can hold the private and public keys of the user, the user's digital certificate, and possibly a copy of the CA's public key. The use of this card shall additionally be secured by, e.g. at least use of a PIN[7], increasing the security of the system by requiring the user to possess the card and to know how to access it. Only a proprietary card reader can be

[7] Personal Identification Number

used to retrieve information from the smart card, therefore minimizing the risks of unwanted abuse.

3.5.3 X.509 Standard

The ITU-T Recommendation X.509 (formerly CCITT X.509) defines a framework for the provision of authentication services, under a central control paradigm represented by a *Directory*. The directory is, in effect, a repository or distributed set of servers that maintains a database of information about users. It describes two levels of authentication: simple authentication, using a password as a verification of claimed identity; and strong authentication, involving credentials formed by using cryptographic techniques.

The X.509 standard defines what fields are to be contained in a certificate, and describes how to write it down (the data format). All X.509 certificates have the following data, in addition to the signature:

- Version

This identifies which version of the X.509 standard applies to this certificate, which affects what information can be specified in it. Thus far, three versions are defined.

- Serial Number

The entity that created the certificate is responsible for assigning it a serial number to distinguish it from other certificates it issues. This information is used in numerous ways, for example when a certificate is revoked its serial number is placed in a Certificate Revocation List (CRL).

- Signature Algorithm Identifier

This identifies the algorithm used by the CA to sign the certificate.

- Issuer Name

The X.500 name of the entity that signed the certificate. This is normally a CA. Using this certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as root or top-level CA certificates, the issuer signs its own certificate.)

- Validity Period

Each certificate is valid only for a limited amount of time. This period is described by a start date and time and an end date and time, and can be as short as a few seconds or almost as long as a century. The validity period chosen depends on a number of factors, such as the strength of the private key used to sign the certificate or the amount one is willing to pay for a certificate. This is the expected period that entities can rely on the public value, if the associated private key has not been compromised.

- Subject Name

The name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across the Internet. This is the Distinguished Name (DN) of the entity, for example,

CN=Java Duke, OU=Java Software Division, O=Emorphia, C=UK

(These refer to the subject's Common Name, Organizational Unit, Organization, and Country.)

- Subject Public Key Information

This is the public key of the entity being named, together with an algorithm identifier which specifies which public key crypto system this key belongs to and any associated key parameters.

X.509 Version 1 has been available since 1988, is widely deployed, and is the most generic.

X.509 Version 2 introduced the concept of subject and issuer unique identifiers to handle the possibility of reuse of subject and/or issuer names over time. Most certificate profile documents strongly recommend that names not be reused, and that certificates should not make use of unique identifiers. Version 2 certificates are not widely used.

X.509 Version 3 is the most recent (1996) and supports the notion of extensions, whereby anyone can define an extension and include it in the certificate. Some common extensions in use today are:

1. KeyUsage (limits the use of the keys to particular purposes such as *signing-only*)

2. AlternativeNames (allows other identities to also be associated with this public key, e.g. DNS names, Email addresses, IP addresses).

Extensions can be marked critical to indicate that the extension should be checked and enforced/used. For example, if a certificate has the KeyUsage extension marked critical and set to *keyCertSign* then if this certificate is presented during secure communication, it should be rejected, as the certificate extension indicates that the associated private key should only be used for signing certificates and not for misuse.

All the data in a certificate is encoded using two related standards called ASN.1/DER. Abstract Syntax Notation 1 describes data. The Definite Encoding Rules describe a single way to store and transfer that data.

Technologies that rely on X.509 certificates include various code-signing schemes, such as signed Java ARchives, and Microsoft Authenticode, secure E-Mail standards, such as PEM and S/MIME and e-commerce protocols, such as SET and SSL.

The IETF[8] PKIX[9] working group is in the process of defining standards for the Internet Public Key Infrastructure including its X.509 Certificate and CRL Profile. PKIX drafting progress can be monitored from [18].

[8] Internet Engineering Task Force
[9] Public-Key Infrastructure (X.509)

3.6 Symmetrical and Asymmetrical Cryptographic Techniques Combined

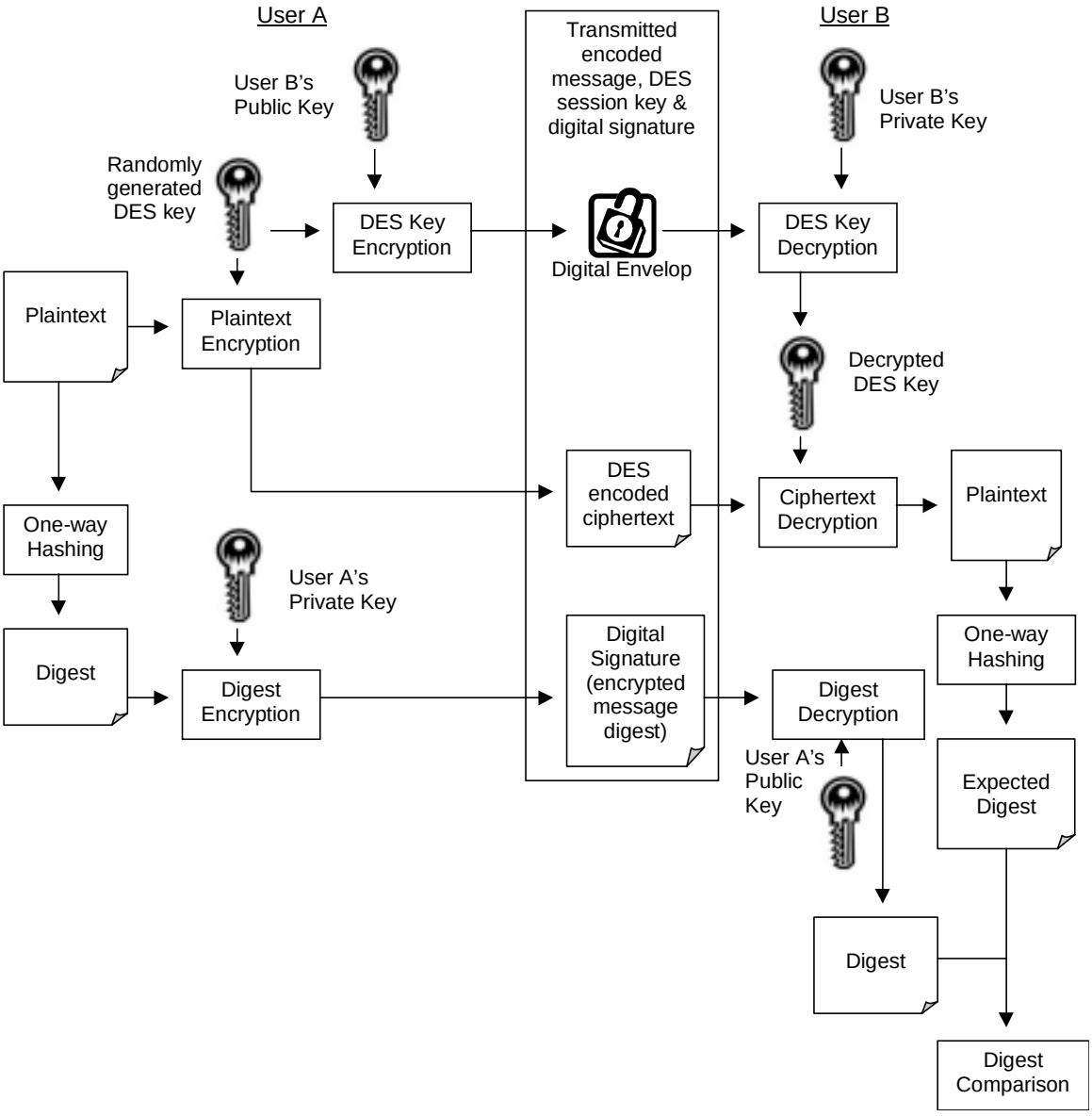

Figure 6: Encryption Techniques Ensuring Message Authenticity, Integrity and Confidentiality

It is clear that the public key cryptographic approach has its biggest drawback in the lack of speed [19]. This has given rise to the wide use of a combination of symmetric key technology, such as DES, with public key cryptography, such as RSA. RSA is not an alternative or replacement for DES, rather it supplements DES (or any other fast bulk encryption cipher) and is used together with DES to provide a more efficient, effective and secure solution to real-life application, especially in the electronic marketplace. The cryptographic protocols that make heavy use of the working principles of these various cryptographic techniques are SSL and SET.

RSA allows two important functions not provided by DES: secure key exchange without prior exchange of secrets, and digital signatures for the purpose of authentication. For encrypting messages, RSA and DES are usually combined as follows (Figure 6): User A first encrypts the message with a randomly generated DES key, and prior to being sent over an insecure communications channel, the DES key is encrypted with User B's RSA public key. Together, the DES-encrypted message and the RSA-encrypted DES key are sent. This protocol is sometimes known as an RSA *digital envelope*.

User A also wishes to sign the message sent to User B. A proceeds by performing a hash function on the message to create a message digest, which serves as a *digital fingerprint* of the message. A then encrypts the message digest with A's own RSA private key; this is the digital signature, which A sends to B along with the message itself. B, upon receiving the message and signature, decrypts the signature with A's public key to recover the message digest. A then hashes the message with the same hash function A used and compares the result to the message digest decrypted from the signature. If they are identical, the signature has been successfully verified and B can be confident that the message did indeed come from A. If, however, the message digests are dissimilar, then the message either originated elsewhere or was altered after it was signed, and B rejects the message.

It must be infeasible for anyone to either find a message that hashes to a given value or to find two messages that hash to the same value. If either were feasible, an intruder could attach a false message onto A's signature. Hash functions such as MD4 and MD5 have been designed specifically to have the property that finding a match is infeasible, and are therefore considered suitable for use in cryptography.

For authentication, the roles of the public and private keys are converse to their roles in encryption, where the public key is used to encrypt and the private key to decrypt. In practice, the public exponent is usually much smaller than the private exponent; this means that the verification of a signature is faster than the signing. This is desirable because a message or document will only be signed by an individual once, but the signature may be verified many times.

In certain situations, RSA is not necessary and DES alone is sufficient. This includes multi-user environments where secure DES-key agreement can take place, for example by the two parties meeting in private. Also, RSA is usually not necessary in a single-user environment; for example, if you want to keep your personal files encrypted, just do so with DES using your personal password as the DES key. RSA, and public-key cryptography in general, is best suited for a multi-user environment, such as an electronic marketplace.

3.6.1 Key Distribution

Figure 6 deals only with the distributions of DES session key and the message itself but does not address the public key distribution problem. Distribution of public keys can be securely conducted with the use of digital certificates which can guarantee trustworthy exchange of owners' public keys through the involvement of a CA in a Public Key Infrastructure (PKI), who plays the role as an independent arbitrator (discussed in 3.5). One or more certificates usually accompany a digital signature. Each certificate is a signed document attesting to the identity and public key of the person signing the message. Its purpose is to prevent someone from impersonating someone else, using a phony key pair. If a certificate is present, the recipient (or a third party) can check the authenticity of the public key, assuming the CA's public key is itself trusted, as discussed in 3.4.1.

However, certificates can also be revoked, either because the user's public key has been compromised, the CA's key has been compromised, or because the CA no longer wants to certify the user. Each CA must maintain a list of all revoked but not expired certificates. When a user receives a new certificate, the person should check the database of revoked keys, called the Certificate Revocation List (CRL) on the network to see if it has been revoked.

4 Real World Cryptographic Applications

4.1 Internet Security Approaches

Originated as an experiment in internetworking, the TCP/IP were never designed to run something as vast as the Internet or were going to be entrusted with confidential data. Consequently, security measures are being implemented as additional features of the existing network. During its tremendous growth period, the Internet has become a vulnerable target that has attracted a fair number of intruders who try to exploit its vulnerability for their own advantage.

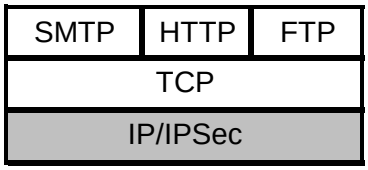

Figure 7

Therefore, much effort has been channeled into developing new mechanisms that will complement and enhance the security aspect of the TCP/IP protocol stack. One way is to use IP Security (IPSec). The primary advantage of using IPSec is that it is transparent to end-users and applications as a general-purpose solution. Additionally, IPSec includes a filtering capability so that only selected traffic need incur the overhead of IPSec processing (Figure 7).

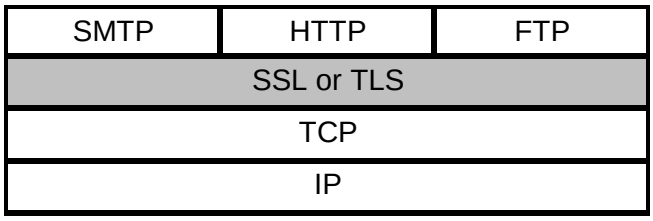

Figure 8

Another relatively general-purpose solution is to implement security above the TCP. The foremost choice of this approach is SSL (Figure 8). SSL could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages. For example, Netscape and Mircosoft Explorer browsers come equipped with SSL support, and most Web servers have implemented the protocol. Microsoft has also developed a similar protocol known as Transport Layer Security (TLS).

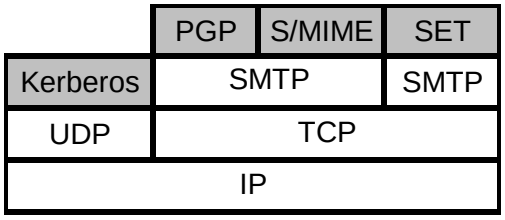

Figure 9

Yet another approach is targeted at the application level whereby security services are embedded within the particular application (Figure 9). The advantage of this approach is that the service can be tailored to the specific needs of a given application. In the context of Internet security, SET is a standard that has been accorded wide industry support. In the next sections, we shall consider SSL and SET as the possible solutions to meet the business requirements of a secure electronic marketplace.

4.2 Secure Socket Layer

Netscape introduced SSL in 1994 with the first version of the Netscape Navigator browser and has since paved the way for online information security. While SSL is not the only viable method (others include SET and S-HTTP[10]) for providing secure transactions on the Internet, it is the most popular [19].

The primary goal of the SSL protocol is to provide privacy and reliability between two communicating applications. The protocol is composed of two layers. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the SSL Record Protocol (Figure 10). The SSL Record Protocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the HTTP, provides the transfer service for Web client/server interaction. Other higher level protocols that can reside on top of SSL include telnet, SMTP, ftp, gopher etc. Three higher level protocols are also defined as part of SLL: the *Handshake Protocol*, the *Change Cipher Spec Protocol*, and the *Alert Protocol*.

At each layer, messages may include fields for length, description, and content. SSL takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled, then delivered to higher level clients.

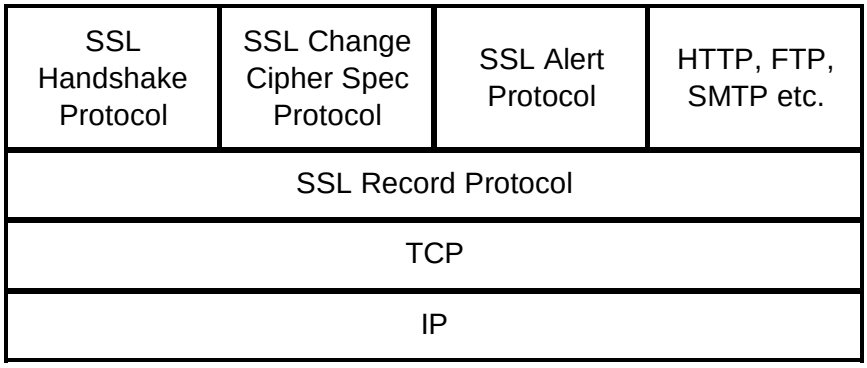

Figure 10: SSL Protocol Stack

The SSL protocol provides connection security that has three basic properties:

- Authentication

The server's identity can be authenticated using asymmetric, or public key cryptography (e.g., RSA, DSS etc.). If visitors to a site use personal digital certificates, the server can instantly recognize them, facilitating instant log-in and preventing later repudiation of the Web transaction. This feature is optional due partly to the unpopularity of digital certificate ownership among the Web users.

[10] Secure Hypertext Transfer Protocol is developed by Terisa Systems, Inc. to provide secure communication mechanisms between an HTTP client-server pair

- Message Integrity

SSL preserves the integrity of every transaction, generating a warning if so much as one character of information is tampered between a server and a customer's browser. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g., SHA, MD5, etc.) are used for MAC computations.

- Confidentiality

Users are assured that no unauthorized third party has intercepted any useful data, like account numbers or credit card numbers, en rout to the intended destination. Encryption is used after an initial handshake to define a secret key. Symmetric cryptography is used for data encryption (e.g., DES, RC4, 3DES etc.)

4.2.1 Weaknesses of SSL

Probably the biggest flaw of SSL is the lack of security and control over the customers' payment information. When data arrives at the merchant's web site, all the information is decrypted and whether or not such information is stored in a secure format is the responsibility of the merchant, the customer has no control over the security of their information. The customers have no assurance that the merchant is authorized to accept credit card payment, neither do they know for certain that the credit card information will be guarded securely.

On the merchant's side, the merchant has no proof that the customer is the true owner of the credit card. This is a risk that the merchant and the credit card vendor assume and factor in to their cost of doing business. This risk increases with the purchases of *soft goods* such as software and games where the purchase is delivered over the Internet.

Credit card transactions via SSL do not guarantee that the transaction/payment can be fulfilled successfully. Even if money has been transferred to a merchant's account, the cardholders are able to enforce a charge back [1]. This is because Internet credit card transactions are classified as Card-Not-Present transactions, therefore merchants are liable for losses, even when the bank has authorized the transactions. However, SET does not have this problem.

4.3 Secure Electronic Transaction

Secure Electronic Transactions (SET) is a cryptographic protocol jointly developed by Visa, MasterCard, Netscape, and Microsoft. Unlike SSL, which is a general purpose system for encrypting communications, SET is only used for secure credit and debit card transactions between customers and merchants. SET is designed to add confidence to the payment process by ensuring that merchants are authorized acceptors of the payment card, thereby eliminating the whole category of merchant fraud, and also by ensuring that the customer is an authorized user of the payment card.

At a low level, the SET protocol provides all the features that are supported by the SSL protocol, which include authentication, message integrity and confidentiality. Additionally, SET prevents the merchant from learning the cardholder's credit card number; this is only provided to the issuing bank. Since SET calls for the mutual

authentication of two communicating parties, the target of non-repudiation is also achieved.

However, in order to implement SET into the payment infrastructure, there is a need for additional hardware support. A SET transaction is also fairly complex so some performance slowdown on the web server is unavoidable.

A SET system includes the following participating elements:

- The Card Holder

A cardholder is an authorized holder of a payment card that has been issued by an issuer.

- The Merchant Account

Before credit card transactions can be accepted, first there is a need to set up a merchant account with a merchant bank. Merchant status requires that a business operator deal with a sponsoring bank to help capture and deposit funds.

- The Payment Processor

Behind the scenes, credit card transactions are fairly complex. In every transaction, there are multiple independent groups participating, including the online merchant, the customer, the merchant's bank, the bank that issued the credit card and the online processor or clearinghouse that manages the whole operation. Before the customers' money can make it into the merchant's bank account, there are a series of steps a transaction must go through, including:

1. Authentication and Purchase Request

Verifies that the credit card has valid numbers, has been officially issued, and has not been reported stolen.

2. Payment Authorization

Checks to make sure adequate funds are available to make the purchase. If they are, the funds are *reserved* or *parked* for the merchant.

3. Payment Capture/Settlement

Once the products have been shipped, the banks need to know so that they can release the parked or reserved funds to the merchant.

Having a merchant account is not enough. All a merchant account does is enable the seller to accept credit cards. To process these cards, a payment processor or transaction service is needed. Also known as *gateways*, these infrastructures make e-commerce feasible.

- The Merchant's Bank/Acquirer

The merchant's bank, also know as an *acquirer* establishes an account with the merchant and processes payment card authorizations and payments. The acquirer

provides authorization to the merchant and keeps tracks of the purchase limits of a cardholder.

- Issuer

An issuer is usually a financial institution that provides the cardholder with the payment card.

- Certificate Authority (CA)

An agent of one or more payment card brands belonged to the larger PKI that provide for the creation and distribution of electronic certificates for cardholders, merchants, and payment gateways.

- Payment Card Brand's Financial Network

The existing private network operated by a payment card brand that links Acquirers and Issuers.

The next sections describe the working implementation of a SET system.

4.3.1 Purchase Request

The SET protocol is invoked after the cardholder has completed browsing, selection and ordering. Before this flow begins, the cardholder will have been presented with a completed order form and approved its contents. In addition, the cardholder will have selected a bankcard as the means of payment. In order to send SET messages to a merchant, the cardholder must have a copy of the merchant public key as well as the Payment Gateway's public key. The SET order process is started when the cardholder requests a copy of the merchant's and gateway's certificates. The message from the cardholder indicates which bankcard brand will be used for the transaction.

When the merchant receives the request, it assigns a unique transaction identifier to the message. It then transmits the merchant and gateway certificates that correspond to the bankcard brand indicated by the cardholder along with the transaction identifier to the cardholder.

The cardholder software verifies the merchant and gateway certificates by traversing the certificate chain to the root CA. The software must hold these certificates to use later during the ordering process. The cardholder software creates the Order Information (OI) and Payment Instructions (PI). The software includes the cardholder signature certificate with the OI; it also includes the cardholder signature certificate with the PI. The software places the transaction identifier assigned by the merchant in the OI and the PI; this identifier will be used by the Payment Gateway to link the OI and the PI together when the merchant requests authorization.

The cardholder software generates a dual signature for the OI and the PI by computing the message digests of both, concatenating the two digests, computing the message digest of the result and encrypting the resulting dual hash using the cardholder private signature key. The message digests of the OI and the PI are sent along with the dual signature.

Next the software generates a random symmetric encryption key and uses it to encrypt the dual signed PI. The software then encrypts the cardholder account number as well as the random symmetric key used to encrypt the PI into a digital envelope using the Payment Gateway's public key. Hence, only the acquirer who runs the Payment Gateway can decrypt this digital envelope. Finally, the software transmits a message consisting of the OI and the PI to the merchant.

When the merchant software receives the order, it verifies the cardholder signature certificate by traversing the certificate chain to the root CA. Next the merchant software verifies cardholder dual signature on OI by decrypting it with the cardholder's public key and comparing the result with the newly generated hash of the concatenation of the message digests of the OI and the PI (included with the OI). This will ensure that the order has not been tampered with in transit. The merchant software then processes the order including the payment authorization described in Section 4.3.2.

After the OI has been processed, the merchant software generates and digitally signs a purchase response message, which includes the merchant signature certificate and indicates that the cardholder's order has been received by the merchant. The response is then transmitted to the cardholder. If the authorization response (see Section 4.3.2) indicates that the transaction was approved, the merchant will ship the goods or perform the services indicated in the order.

When the cardholder software receives the purchase response message from the merchant, it verifies the merchant signature certificate by traversing the certificate chain to the root CA. It uses the merchant public signature key to check the merchant's digital signature. Finally, it takes some action based on the contents of the response message, such as displaying a message to the cardholder or updating a database with the status of the order.

4.3.2 Payment Authorization

During the processing of an order from a cardholder (see Section 4.3.1), the merchant will authorize the transaction. The merchant software generates and digitally signs an authorization request, which includes the amount to be authorized, the transaction identifier from the OI and other information about the transaction. The request is then encrypted using a new randomly generated symmetric key, which in turn is encrypted using the public key of the Payment Gateway. (This is the same key the cardholder used to encrypt the digital envelope of the payment instructions.) The authorization request and the cardholder payment instructions are then transmitted to the Payment Gateway.

When the Payment Gateway receives the authorization request, it decrypts the digital envelope of the authorization request to obtain the symmetric encryption key. It uses the symmetric key to decrypt the request. It then verifies the merchant signature certificate by traversing the certificate chain to the root CA; it also verifies that the certificate has not expired. It uses the merchant public signature key to ensure the request was signed using the merchant private signature key.

Next the Payment Gateway decrypts the digital envelope of the Payment Instructions to obtain the symmetric encryption key and the account information. It uses the symmetric key to decrypt the PI. It then verifies the cardholder signature certificate by traversing the certificate chain to the root; it also verifies that the certificate has not expired. Next it uses the cardholder public signature key and the message digest of the OI (included in

the PI) to check the digital signature to ensure that the PI has not been tampered with in transit and that it was signed using the cardholder private signature key.

Next, the Payment Gateway verifies that the transaction identifier received from the merchant matches the one in the cardholder Payment Instructions. The Payment Gateway then formats and sends an authorization request to the cardholder's financial institution (Issuer) via an existing bankcard payment system.

Upon receiving an authorization response from the Issuer, the Payment Gateway generates and digitally signs an authorization response message, which includes the Issuer's response and a copy of the Payment Gateway signature certificate. The response also includes an optional capture token with information the Payment Gateway will need to process a capture request (see Section 4.3.3). The capture token is optional and is only included if required by the Acquirer.

The response is then encrypted using a new randomly generated symmetric key, which in turn is encrypted using the merchant public key. The response is then transmitted to the merchant.

When the merchant software receives the authorization response message from the Payment Gateway, it decrypts the digital envelope to obtain the symmetric encryption key. It uses the symmetric key to decrypt the response message. It then verifies the Payment Gateway signature certificate by traversing the certificate chain to the root CA. It uses the Payment Gateway public signature key to check the Payment Gateway digital signature. The merchant software will store the authorization response and the capture token to be used when requesting payment through a capture request (see Section 4.3.3). The merchant then completes processing of the cardholder's order (see Section 4.3.1).

4.3.3 Payment Capture

After completing the processing of an order from a cardholder (see Section 4.3.1), the merchant will request payment. There will often be a significant time lapse between the message requesting authorization and the message requesting payment. The merchant software generates and digitally signs a capture request, which includes the final amount of the transaction, the transaction identifier from the OI and other information about the transaction. The request is then encrypted using a new randomly generated symmetric key, which in turn is encrypted using the public key of the Payment Gateway. The capture request and optionally the capture token if one was included in the authorization response (see Section 4.3.2) are then transmitted to the Payment Gateway.

When the Payment Gateway receives the capture request, it decrypts the digital envelope of the capture request to obtain the symmetric encryption key. It uses the symmetric key to decrypt the request. It then uses the merchant public signature key to ensure the request was signed using the merchant private signature key. The Payment Gateway decrypts the capture token (if present) and then uses the information from the capture request and the capture token to format a clearing request, which it sends to the Issuer via a existing bankcard payment system.

The Payment Gateway then generates and digitally signs a capture response message, which includes a copy of the Payment Gateway signature certificate. The response is then encrypted using a new randomly generated symmetric key, which in turn is

encrypted using the merchant public key. The response is then transmitted to the merchant.

When the merchant software receives the capture response message from the Payment Gateway, it decrypts the digital envelope to obtain the symmetric encryption key. It uses the symmetric key to decrypt the response message. It then verifies the Payment Gateway signature certificate by traversing the certificate chain to the root CA. It uses the Payment Gateway public signature key to check the Payment Gateway digital signature.

The merchant software will store the capture response to be used for reconciliation with payment received from the Acquirer.

4.4 Identrus

Identrus was launched in April 1999 by ABN AMRO, Bank of America, Deutsche Bank, Barclays, Chase Manhanttan, Citigroup and Hypo Vereinsbank as a for-profit limited liability company [23]. Its mission is to help organizations surmount the final obstacle preventing B2B electronic commerce from thriving. Through a relationship with participating financial institutions (also known as *Identrus Certificate Authority*), companies will be able to use the Internet to conclusively identify trading partners and conduct trusted B2B commerce with any other participant in the Identrus system. All these companies need is just one *Identrus Global ID* for all their B2b activities.

Identrus helps businesses to actively manage their electronic commerce risks through trusted relationships with their financial institutions, which is not unlike SET. Specifically, Identrus supplies the legal mechanisms and technology to let Internet trading partners trust in one another's identities. Similar to SET, the driving force behind Identrus is the Public Key Infrastructure that provides secure Internet-based transaction from sourcing through negotiation, delivery and payment.

For a typical financial transaction in an Identrus system, a merchant would request its financial institution to validate the Identrus Global ID of the buyer. The merchant's financial institution would electronically contact the buyer's financial institution, which in turn would confirm the identity of its customer, i.e. the buyer. Identrus will validate the respective financial institution's identity as part of the process. The party relying on an Identrus Global ID may obtain an identity warranty to manage any residual risk through its financial institution. An identity warranty is a positive affirmation by a financial institution to stand behind its Identrus Global IDs by offering financial recourse should something go wrong with the identity. This provides replying parties additional flexibility when managing risks associated with evaluating and accepting electronic identities in electronic commerce.

Finally, feasibility analysis and comparison are not possible for Identrus because it is the first and only company injecting high trust into B2B electronic commerce. Moreover, Identrus is new in the electronic marketplace. Having made its debut only a mere two years ago, there is very little application of this new system.

5. Analysis

5.1 Performance Concerns

Using SET, only the sensitive information in the transaction such as name, address, credit card details, etc.) is encrypted. When the customer visits the web site, the pages are not encrypted as they are downloaded onto the customer's PC, so the web site designer is free to use graphics more liberally as there is no encryption performance hit.

As for SSL, during a typical session, all communication over the Internet is encrypted, including graphics and images used within a web page, in addition to the transaction information. This can have an impact on the overall performance of the protocol. Nowadays merchants are forced to use minimal graphics on their SSL-protected web server. Merchants who ignore this fact could lose customers because their complex web pages are too slow to download.

A downside of both of these protocols is that they both require the use of cryptographic algorithms that place significant loads on the computer systems involved in the payment transactions. SSL has a lower impact on the e-commerce server but does less to eliminate the security risk. SET has a higher performance impact, but allows for a much more secure payment transaction.

However, the are available technologies that will improve the performance of the web server used in the payment infrastructure. These technologies include:

- Cryptographic accelerators

Cryptographic accelerators are special-purpose dedicated hardware units designed to off-load cryptographic operations from the CPU. Unlike 32-bit CPUs, which are inefficient at 1024-bit arithmetic, the processors in cryptographic accelerator hardware are designed specifically for long number arithmetic. The correct analogy is to think of cryptographic accelerators like the graphics accelerators used in nearly every PC today. Graphics accelerators perform high-speed functions that offload the CPU and result in a system with substantially increased overall performance. Similarly, in systems with a substantial cryptographic load, cryptographic accelerators offload the CPU for much better overall system performance.

- Symmetric multiprocessing (SMP) CPU scaling

In a SMP system, the operating system (OS) allocates individual CPU functionality to application and other processes as required. As more processors are added to a system, it is possible for the OS to dedicate individual processors to supporting cryptography.

- Clustering

In most large sites, the application load is shared between multiple multiprocessor CPUs, linked together into a cluster so that in case of an equipment failure, the other systems in the cluster will absorb the load until the failing system can be brought back

online. In these environments, the additional systems provide the ability to spread the transaction load over multiple systems in the same cluster.

- Elliptical curve cryptography (ECC)

A more recent development in the field of public key systems in based on the use of elliptic curves. The use of elliptic curves was first proposed by Miller in 1986, and followed by Koblitz in 1987 [20]. The principal feature of ECC compared to RSA is that it appears to offer equal security for a far smaller bit size, thereby reducing processing overhead. The effect of this is to greatly increase the speed with which cryptographic operations can be performed. This technology has not been widely deployed, and as such is not as well proven as the other technologies included here.

Finally, looking at server performance as a static situation is clearly the wrong thing to do since technology is continuously improving. For example, Moore's law tells us that the performance of the CPUs used in the servers will double in performance every 18 months. Therefore, the performance impact on the transaction web servers should not pose a big problem in the future.

5.2 Price

According to the SET Comparative Performance Analysis conducted by Gartner Group in 1998, the cost of additional hardware support required to support SET is small in comparison to SSL. In fact, the margin of difference is almost negligible if future advancement in technologies is taken into accounts. Their findings are as follow [21]:

- For the low and medium e-commerce applications, there is no additional server cost to support SET over SSL. The performance of the servers in these price ranges, at the anticipated loading, is more than adequate today and through the forecast period.

- For the large e-commerce server application, supporting SET requires additional hardware acceleration in the medium term with a 5 percent to 6 percent difference in server cost.

- For the small payment gateway application, hardware acceleration is required in the short term, but can be phased out as servers improve in performance and if other performance improvements such as elliptical-curve cryptography (ECC) become available.

- It is anticipated that large payment gateway applications will always be based on clustered systems for reasons of robustness and reliability. In these environments the difference in cost to support SET over SSL will be covered by the investment in additional clustered systems. In the short term, because it is hypothetically possible with significant hardware acceleration to use a single system to support SSL, the cost of requiring a clustered system to support SET would appear to double the cost compared to a single SSL system. However, this is not a recommended configuration. In the medium term, this payment gateway application will require additional hardware acceleration to support SET resulting in a 5 percent increase in server cost.

In short, the potential benefit to all investors in e-commerce, the end-users (customers) included, would seem to easily outweigh the extra investment in SET.

5.3 Flexibility

Version 1.0 of the SET specification was published May 31, 1997. Future releases will be backwards compatible, so there will be no need to upgrade unless a new business functionality is determined to be necessary by the vendors. Enhancements to the protocols are always being considered. The SETCo advisors groups, who form the monitoring body of the SET protocol constantly review the requirements, define extensions to the protocol and approve those extensions as they are completed. Additionally, there is a very active interoperability effort by the vendors to work on overcoming incompatibilities between their products, thus assuring a high level of flexibility and interoperability of the SET protocol, as well as the compliant software applications.

5.4 Availability

For a merchant to take advantage of SET, the merchant need to purchase a desired SET compliant software application and obtain digital certificates to set up a SET online merchant site. There are already many SET compliant products in the market. Software that has passed SET Compliance Testing has earned the right to use the SET Mark on their web pages if the functionality is turned on, but there may be cases where an unlicensed use of the SET mark is being displayed on a site. Therefore, the best way to know is by checking the SETCO official web site where a Vendor Status Matrix and Derived Product matrix are maintained. The matrixes are lists of approved vendors that offer SET compliant solutions. Therefore, once a product has been procured from the right vendor, a merchant needs only to obtain the necessary certificates from the financial institutions that participate in the scheme.

It is a fact that SET relies heavily on the Public Key Infrastructure in order to function seamlessly. In today's electronic marketplace there is already a growing number of commercial PKI service providers including VeriSign (USA), Global Sign (Belgium), Telesec (1st CA compliant to German Digital Signature act) and TC Trustcenter (Germany). Other PKI product vendors include Baltimore, Entrust, IBM, id2, RSA, Spyrus, SSE etc.

6. Summary

6.1 Conclusion

By delving into the underlying principles of cryptography and its application through the SET and SSL protocols, we found the solutions to the B2C payment problem over the Internet. The SET protocol especially has ensured four vital security objectives for the electronic marketplace, i.e. data confidentiality, integrity, authenticity and non-repudiation of transaction while incurring little or no significant decrease in server access time. SET could well be the preferred payment solution in the future electronic marketplace. Already SET is showing sign of gradual adoptions by many financial institutions and vendors. However, a successful worldwide deployment of the SET infrastructure still hinges heavily on several factors:

a. Deployment of PKI with worldwide interoperability and integration

b. High quality SET-compliant application from software vendors

c. Supply of bandwidth that matches the increasing demand

Furthermore, it is my opinion that a successful distribution of smart card technology can also accelerate the deployment of PKI (the many benefits of smart card technology were discussed in 3.5.2). There is also a need for proper legal framework (e.g. digital signature legislation, industry regulations) in countries aspiring to embrace the digital age.

In the area of modern B2B electronic commerce, the Identrus infrastructure seems like the only logical and viable solution. However, the real business benefit of using Identrus-enabled security waits to be seen. The question is how to deploy it cheaply and efficiently. After all, many companies have been conducting B2B transactions through EDI for many years, so there is a real need for the cost-effectiveness of Identrus to be displayed. Furthermore, many Identrus-based solutions are still in prototypes and widespread adoption of this system will take time.

Bibliography

[1] Amor, D., *The E-Business (R)evolution*, New Jersey: Prentice Hall, 1999.

[2] Stalling, W., *High-Speed Networks, TCP/IP and ATM Design Principles*, New Jersey: Prentice Hall, 1998.

[3] The DSL Forum, General Introduction to Copper Access Technologies, URL http://www.adsl.com/general_tutorial.html

[4] Zheng, H., Liu, K.J. R., "Multimedia Service over Digital Subscriber Lines," IEEE Signal Processing Magazine, vol. 17, July 2000, pp. 47-48.

[5] K. Maxwell, "Asymmetric Digital Subscriber Line: Interim Technology for the Next Forty Years," IEEE Communications Magazine, vol. 34, Oct. 1996, pp. 101-103.

[6] Dutta-Roy, A., "A second wind for wiring," IEEE Spectrum, vol. 36, September 1999, pp. 52- 60.

[7] Wireless Developer Network, URL: http://www.wirelessdev.com

[8] Abu El-Ata, M. A., "Evolution of Mobile Cellular Communication Systems- The Journey to UMTS," A paper presented at the 17th National Radio Science Conference, NRSC'2000, Feb. 22-25, 2000.

[9] Gudding, H., "Capacity Analysis of GPRS White Paper," Master Thesis at the Norwegian University of Science and Technology, Nov. 2000.

[10] Ahuja, V., *Secure Commerce on the Internet*, Boston: AP Professional, 1995.

[11] Jackson, K. M., & Hruska, J. (Eds.)., *Computer Security Reference Book*, Florida: CRC Press, 1992.

[12] Stallings, W., *Cryptography and Network Security: Principles and Practice,* New Jersey: Prentice Hall, 1999

[13] Roback, E. & Dworkin, M., "First Advanced Encryption Standard (AES) Candidate Conference," *CryptoBytes*, vol. 4, pp.6-14, Winter 1999.

[14] Stein, L. D., *Web Security: A Step-by-Step Reference Guide,* Reading, Massachusetts: Addison-Wesley, 1998.

[15] Greenstein, M., & Feinman, T. M., *Electronic commerce: Security, Risk Management and Control,* New York: Irwin McGraw-Hill, 1999

[16] Keen, P. et al., *Electronic Commerce Relationships-Trust by Design*, New Jersey: Prentice Hall PTR, 1999.

[17] Knudsen, J., *JAVA Cryptography*. Cambridge: O'reilly, 1998.

[18] IMC Organization, URL: http://www.imc.org/ietf-pkix/

[19] Vacca, J., *Internet Security: SECRETS*. Foster City, CA: IDG Books,1996.

[20] Lubbe, J. C. A. van der., *Basic Methods of Cryptography,* Cambridge: Cambridge University Press, 1998.

[21] le Tocq, C., Young, S., “SET Comparative Performance Analysis,” *A White Paper from GartnerGroup*, pp.25-26, Nov. 1998.

[22] Rhee, M. Y., *Cryptography and Secure Communications*. Singapore: MacGraw-Hill Book Co., 1994.

[23] Identrus Official Website, URL: http://www.identrus.com/